ELSEVIER

Microelectronics Journal 34 (2003) 975–980

Microelectronics Journal

www.elsevier.com/locate/mejo

An efficient reconfigurable multiplier architecture for Galois field GF(2^m)

P. Kitsos*, G. Theodoridis, O. Koufopavlou

VLSI Design Laboratory, Department of Electrical and Computer Engineering, University of Patras, Rio, Patras 26500, Greece

Received 8 January 2003; revised 26 May 2003; accepted 4 June 2003

Abstract

This paper describes an efficient architecture of a reconfigurable bit-serial polynomial basis multiplier for Galois field GF(2^m), where $1 < m \leq M$. The value m, of the irreducible polynomial degree, can be changed and so, can be configured and programmed. The value of M determines the maximum size that the multiplier can support. The advantages of the proposed architecture are (i) the high order of flexibility, which allows an easy configuration for different field sizes, and (ii) the low hardware complexity, which results in small area. By using the gated clock technique, significant reduction of the total multiplier power consumption is achieved.

Keywords: Galois field; Polynomial multiplication; Bit-serial; Irreducible polynomial; All-one polynomial; Linear Feedback Shift Register; Low power; Cryptography; Elliptic curves

1. Introduction

Arithmetic operations over GF(2^m) have many applications in coding theory [1] and cryptography [2]. As the multiplication is very costly in terms of area and delay, a lot of research has been performed in designing small area and high-speed multipliers [3–5].

Previous published multipliers over GF(2^m) can be classified into three categories: the bit-serial multipliers [3], with O(m) area requirement, the bit-parallel multipliers [4], with O(m^2) area requirement, and the hybrid [5], which are partially bit-serial and partially bit-parallel. Hybrid multipliers are faster than bit-serial ones, while their area is smaller than that of bit-parallel. Another classification can be considered based on the used basis representation, which may be polynomial, normal, dual or digit [6–8]. The multiplier hardware complexity can be reduced if (i) the irreducible polynomial is an All-One Polynomial (AOP) [9] or a trinomial [10], and (ii) a redundant field representation is used [11].

Many of the previous proposed multipliers have fixed field size, and so, if the irreducible polynomial has to change the multiplier must be redesigned [4,9,10]. In the recent years only few fixed field size multipliers were proposed in which the coefficients of the irreducible polynomial can be modified [3]. However, all the above fixed field size multipliers do not work efficiently in applications with variable field size requirements. In these applications the multipliers always performs all the operations, which are needed for the maximum field size calculations. So, in order to improve the system performance in multiplication cases with field size less than the maximum, a proper and flexible design implementation is required. In the past, multipliers with this feature have been proposed in Refs. [5,12,13].

Nowadays wireless devices are widely used. Since power consumption determines the time between two successive recharges of such a device and the battery life as well, the reduction of power dissipation is vital in such devices. The main source of power dissipation in a CMOS circuit is the switching activity of its nodes, which may contribute more than 90% of the total power consumption [14]. However, a lot of the performed circuit node transitions are wasteful regarding the functionality of the circuit. Hence, avoiding the unnecessary and wasteful transitions is a major task in the low power design.

In this paper, a small area reconfigurable architecture for the Most Significant Bit (MSB)-first, bit-serial, polynomial basis multiplier over GF(2^m) is introduced, where $1 < m \leq M$. m is the degree of the irreducible polynomial and it can be easily changed according to the application requirements. M is the maximum degree of the irreducible polynomial.

* Corresponding author. Tel.: +30-2610-997-323; fax: +30-2610-994-798.

E-mail address: pkitsos@ee.upatras.gr (P. Kitsos).

0026-2692/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0026-2692(03)00172-1

Compared with the multipliers in Refs. [3,5,12] the advantages of the proposed architecture are: (i) the high order of flexibility, which allows an easy configuration for different field degree m, and (ii) the low hardware complexity, which results in smaller area. By using the gated clock technique, significant reduction of the total multiplier power consumption is achieved. The proposed multiplier is suitable for elliptic curve applications [15,16], especially in devices with strict area limitations.

The paper is organized as follows: in Section 2 a brief description of the MSB-first, bit-serial, polynomial basis $GF(2^m)$ multiplier is given. In Section 3, the multiplier proposed reconfigurable architecture is presented. Measurements and comparisons with other multipliers are shown in the Section 4. Section 5 concludes the paper.

2. MSB-first bit-serial $GF(2^m)$ multiplier

Two elements, $A(x)$ and $B(x)$, over $GF(2^m)$ can be expressed as polynomials of degree at most $m-1$ over GF(2):

$$A(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0, \quad \text{with } a_i \in GF(2) \quad 0 \le i \le m-1 \tag{1}$$

$$B(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \cdots + b_1x + b_0, \quad \text{with } b_i \in GF(2) \quad 0 \le i \le m-1 \tag{2}$$

We define the field according to $P(x)$:

$$P(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_1x + p_0, \ \text{with } p_i \in GF(2) \quad 0 \le i \le m-1 \tag{3}$$

a m-order irreducible polynomial over GF(2). This polynomial is also irreducible over $GF(2^m)$ [17]. When the coefficients p_i, in the polynomial of Eq. (3), are equal to one, the irreducible polynomial is named AOP [9].

The modulo multiplication $C(x)$ involves the carry-free multiplication between the two polynomials $A(x)$ and $B(x)$, with the interleave reduction modulo $P(x)$:

$$\begin{aligned} C(x) &= (A(x)B(x)) \bmod P(x) = \left(A(x)\sum_{i=0}^{m-1} b_i x^i\right) \bmod P(x) \\ &= \left(\sum_{i=0}^{m-1} (b_i(x^iA(x))\right) \bmod P(x) \\ &= (b_0A(x) + b_1xA(x) + \cdots + b_{m-2}x^{m-2}A(x) \\ &\quad + b_{m-1}x^{m-1}A(x)) \bmod P(x) \end{aligned} \tag{4}$$

Eq. (4) can be implemented using the following iterative algorithm.

2.1. Polynomial multiplication in $GF(2^m)$ algorithm

Step 1: $C^{-1}(x) = 0$

Step 2: for $i = 0$ to $m-1$ do

$C^i(x) = [C^{i-1}(x)x + b_{m-1-i}A(x)] \bmod P(x)$

Using the fact that: $x^m \bmod P(x) = p_{m-1}x^{m-1} + \cdots + p_1x + p_0$ the $C^i(x)$ in the step 2 can be rewritten as

$$\begin{aligned} C^i(x) =& [c^{i-1}_{m-1}(p_{m-1}x^{m-1} + \cdots + p_1x + p_0) + \cdots + c^{i-1}_0 \\ &+ b_{m-1-i}A(x)] \bmod P(x) = (c^{i-1}_{m-1}p_{m-1} + c^{i-1}_{m-2} \\ &+ b_{m-1-i}a_{m-1})x^{m-1} + (c^{i-1}_{m-1}p_{m-2} + c^{i-1}_{m-3} \\ &+ b_{m-1-i}a_{m-2})x^{m-2} + \cdots + (c^{i-1}_{m-1}p_1 + c^{i-1}_0 \\ &+ b_{m-1-i}a_1)x + (c^{i-1}_{m-1}p_0 + b_{m-1-i}a_0) = \sum_{j=0}^{m-1} c^i_j x^j \end{aligned} \tag{5}$$

where

$$c^i_j = c^{i-1}_{m-1}p_j + c^{i-1}_{j-1} + b_{m-1-i}a_j \text{ and } c^{i-1}_{-1} = 0 \tag{6}$$

The block diagram of a bit-serial multiplier architecture is shown in Fig. 1. The main component of the multiplier is a Linear Feedback Shift Register (LFSR). The coefficients of the irreducible polynomial $P(x)$ and the multiplicand polynomial $A(x)$, are stored in $P(i)$ and $A(i)$ registers, respectively. As the coefficients $b(i)$, of multiplier polynomial $B(x)$, are shifted bit-by-bit in the MSB direction, the partial products $b_i^*A(x)$, are produced in the Partial Product component. The LFSR calculates the summary of the $b_i^*A(x)$ and also performs the reduction modulo $P(x)$. After m clock cycles, the multiplication product $C(x)$ is produced. This implementation is a MSB-first version because the MSB is the first bit which enters to the multiplier.

A hardware implementation of the above multiplier has been proposed in Ref. [18] (Fig. 2).

In order to perform the reduction modulo $P(x)$, the non-zero coefficients $p(i)$, of the irreducible polynomial $P(x)$ configures the LFSR, through the upper-series AND gates. As the multiplier polynomial $B(x)$ is shifted bit-by-bit in the MSB direction, the partial products $b_i^*A(x)$ are produced by the lower-series AND gates.

3. Proposed multiplier architecture

The proposed reconfigurable MSB-first multiplier that can be used for variable field degree m is shown in Fig. 3.

The proposed hardware implementation consists of a bit-sliced LFSR and is very similar to the conventional bit-serial multiplier of Fig. 2. It requires M extra demultiplexers and M extra OR gates. Each slice i, consists of two subfield multipliers (AND gates), one subfield adder (XOR gate), one 2-output demultiplexer, one OR gate, and 3 one-bit registers ($P(i)$, $A(i)$ and $D(i)$). The non-zero coefficients $p(i)$, of the irreducible polynomial $P(x)$, configure the LFSR, through the OR and AND gates of the feedback path.

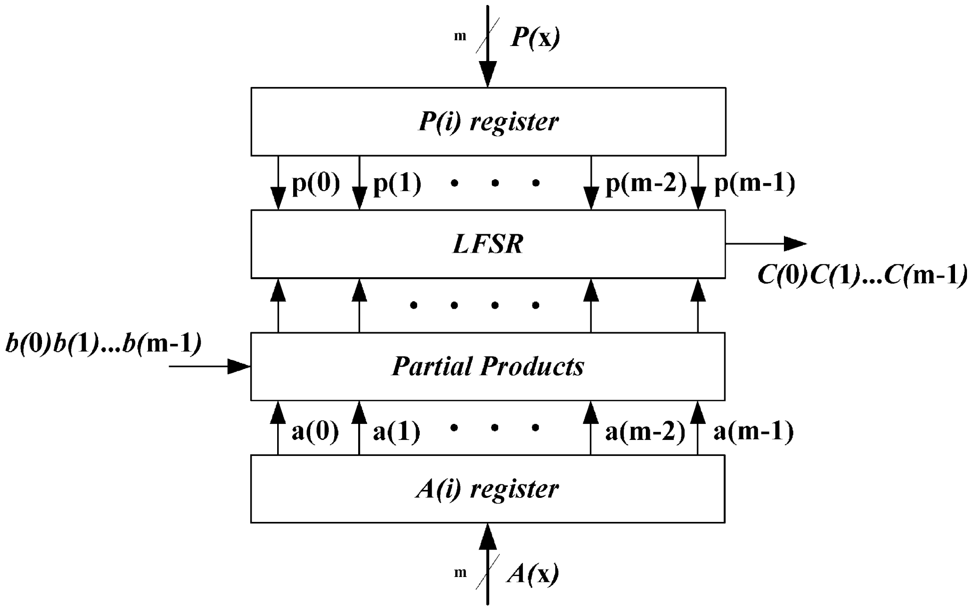

Fig. 1. Block diagram of a bit-serial multiplier.

The maximum value of the field degree is M, and it is determined by the application requirements.

Each coefficient $a(i)$, of the multiplicand polynomial $A(x)$, is stored in $A(i)$ position of the $A(i)$ register, while each coefficient $p(i)$, of the irreducible polynomial $P(x)$, is stored in the $P(i)$ position of the $P(i)$ register. If an irreducible polynomial of degree m, $m < M$ is required, the remaining $P(j)$ and $A(j)$ bits of the registers are filled with zeros, where $m < j \leq M$.

Signal control(i) selects one of the two demultiplexer output. The value of each signal control(i), is defined as:

$$\text{control}(i) = \begin{Bmatrix} 1 & \text{if } i \leq m \\ 0 & \text{if } m < i \leq M \end{Bmatrix} \tag{7}$$

In the positions where control(i) = 1 (out1 is selected) the slice is an active whilst if control(i) = 0 (out2 is selected) the slice is inactive. Inactive means that this slice is not used during multiplication. out2 forces the global feedback path with the proper value through the OR_i gate.

When the application requires multiplications with variable field sizes, the value of the degree m, and the set of coefficients of polynomials $A(x)$, $B(x)$, and $P(x)$ are configured (i.e. programmed) by the Control Unit.

After m clock cycles the right multiplication result is stored in the register $D(i)$. The minimum clock cycle period is determined by the delays of the feedback path and the 2-output demultiplexer. It is equal to $2T_{\text{AND}} + T_{\text{XOR}} + T_{\text{NOT}} + (m+1)T_{\text{OR}}$, where T_{AND}, T_{XOR}, T_{NOT}, and T_{OR} is the delay of the 2-input AND, 3-input XOR, inverter, and 2-input OR gates, respectively.

During computation each one-bit register $D(i)$, is controlled by the corresponding control(i) signal. So, all the one-bit register $D(j)$, are set inactive by discontinue their clock signal. Thus, the unnecessary and wasteful transitions at all the one-bit register $D(j)$, are eliminated. This gated

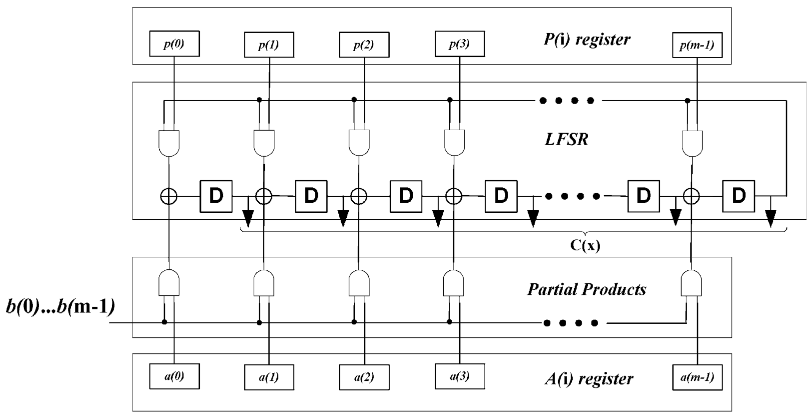

Fig. 2. MSB-first bit-serial multiplier implementation.

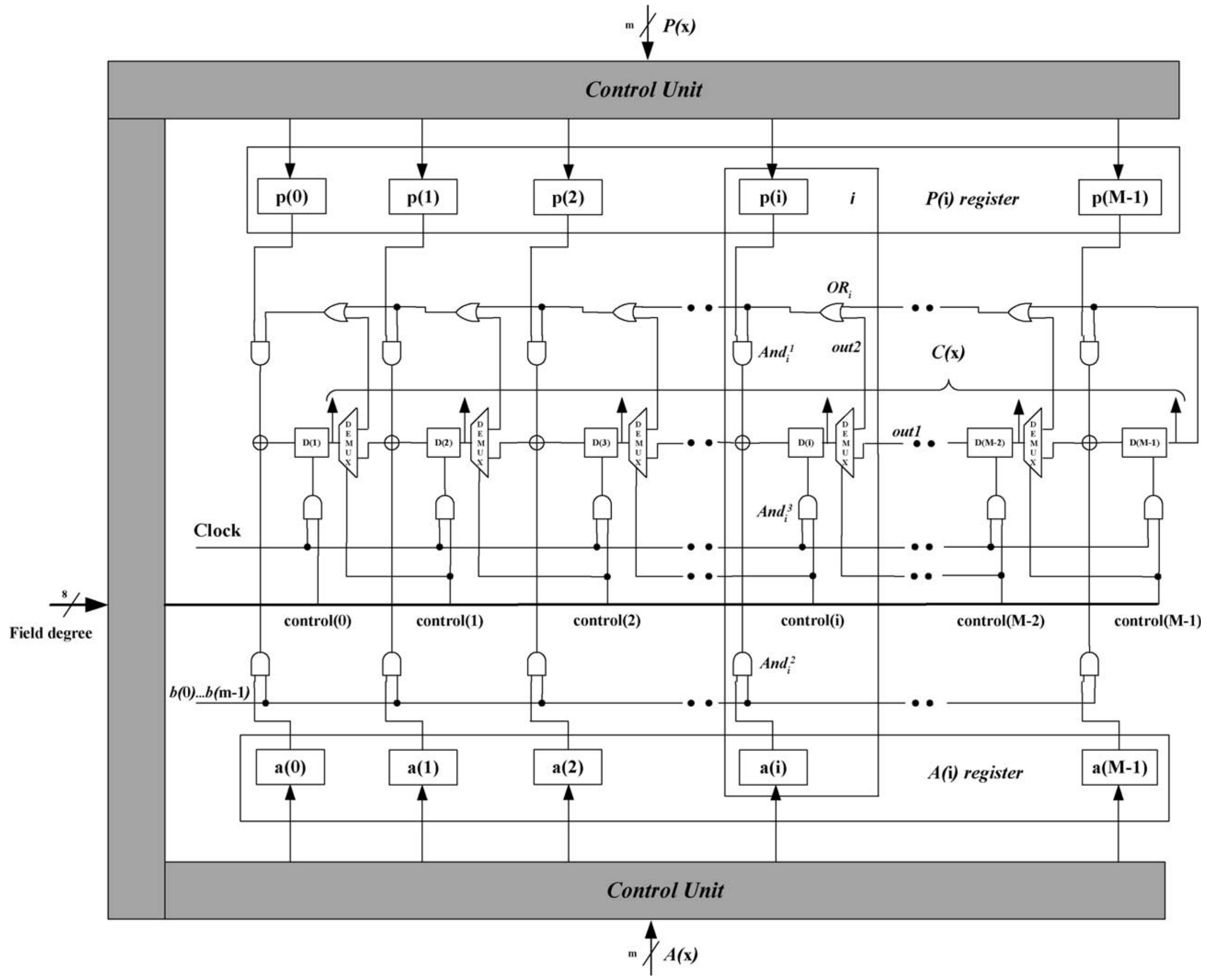

Fig. 3. Proposed reconfigurable bit-serial multiplier.

clock technique results in a significant power dissipation reduction [14].

4. Measurements and comparisons

The area hardware resources and the execution time of the proposed multiplier hardware implementation are shown in Table 1. For comparison, measurements from other designs are also presented.

In Table 1 measurements for the Control Unit area resources are not considered. The bit-serial polynomial basis multiplier suggested in Ref. [3], is based on the Programmable Cellular Automata [19]. It requires less area resources than the proposed implementation, and the critical path is shorter. On the other hand, the main disadvantage of the multiplier implementation in Ref. [3] is that it supports only multiplications with fixed field size.

The implementation in Ref. [5] can operate over variable Galois field size. It uses a representation of the field GF(2^m) as GF($(2^n)^k$) where $m = nk$. This approach processes all component-wise multiplications and additions in bit-parallel in the sub-field GF(2^n) and serial processing by a LFSR for the extension field GF(2^k). The resulting implementation is general in that the value of k can be changed, leading to a limited degree of flexibility. When m is prime and $n > 1$ the multiplication cannot be operated. The usage of composite field is discouraged for elliptic curve cryptosystems. The framework for an attack on elliptic curve cryptosystems that uses composite fields of the form GF($(2^n)^k$) is described in Ref. [20]. The execution time for one multiplication is m/n, lower than the bit-serial one but with higher area complexity.

The implementation proposed in Ref. [12], is a variable Galois field size multiplier. Before performing the multiplication, the operands have to be transformed from polynomial into triangular basis or inverse order. In order

Table 1
Area hardware resources and execution time

Multiplier implementation	Ref. [3]	Ref. [5]	Ref. [12]	Proposed
# AND	$3m$	$(3/4)mn$	$3m$	$2m$
# XOR	m	$m((3/4)n + 3 - 3/n)$	$2m$	m
# REG	m	$m(2 + 1/n)$	m^2	$3m$
# OR	0	0	0	$m - 1$
# (2:1) DEMUX	0	0	0	m
# $(m:1)$ MUX	0	0	m	0
Critical path	$T_{AND} + T_{XOR}$	–	$3m(T_{XOR} + \lceil\log_2 m\rceil (T_{NOT} + T_{AND} + T_{OR}))$	$2T_{AND} + T_{XOR} + T_{NOT} + (m+1)T_{OR}$
# CLK	m	m/n	$3m$	m
Reconfigurable (support any irreducible polynomial)	No	No	Yes	Yes

to support variable field sizes, Serial-In Serial-Out registers, and, m $m:1$ multiplexers are used, resulting in an increase of the clock cycles, which are required for performing multiplication. In addition, the usage of the m $m:1$ multiplexers makes this implementation infeasible for devices with strict area limitations. The multiplication result is computed after $3m$ clock cycles. The critical path is determined by the $m:1$ multiplexer delay and the delay of the 3-input XOR. The total latency is equal to $3m(T_{XOR}+\lceil\log_2 m\rceil(T_{NOT} + T_{AND} + T_{OR}))$.

The implementation proposed in Ref. [13] is highly scalable because a fixed-area multiplier can handle operands of any field size. In this implementation the Montgomery algorithm [21] is used. Moreover, the word size of the processing unit as well as the number of pipeline stages can be selected according to the application requirements. Before performing Montgomery multiplication, the operands must be transformed into the Montgomery domain, while for the completion of multiplication, the final result has to be retransformed to GF(2^m) field. These transformations are accomplished by using pre-computed constants. Additionally, the high number of full adders and registers prohibits the use of this multiplier in applications with strict area limitations.

In applications with limited computational power, a hardware acceleration of the field arithmetic is necessary to reach high performance, especially if the irreducible polynomial degree m is beyond 200. The Elliptic Curve Cryptosystems with key size of 106–210 bits can yield the similar security level with the key size of 512–2048 bits of RSA algorithm. The key sizes are considered to be equivalent strength based on MIPS years needed to recover one key [22].

Table 2
Proposed multiplier performance measurements

Binary field degree (m)	FPGA frequency (MHz)	Multiplication execution time (μs)
106	33.5	3.1
119	30	3.9
132	26	5
158	22.4	7
163	22.1	7.4
174	19.8	8.8
193	18.5	10.4
210	17.1	12.3

In Table 2 measurements for the above mentioned binary field size 2^m are illustrated. The proposed multiplier is implemented in a Field Programmable Gate Array (FPGA). For system synthesis, the LeonardoSpectrum from Mentor Graphics was used. For all the measurements the XILINX FPGA XCV1000E-FG1156-8 device was considered [23]. In this implementation the maximum field degree M is 210. The polynomial degree m, varies from 106 to 210 bits.

The experimental delay measurements of Table 2 are very close to the expected values produced by the theoretical expressions and the data of Table 1. The slight differences between the experimental and the theoretical values are due to for the latest the FPGA internal interconnection wires and buffers delays are not calculated.

5. Conclusion

A reconfigurable bit-serial Galois field multiplier architecture is proposed in this paper. The multiplier is reconfigurable because it can perform for variable Galois field degree m. This multiplier can support any arbitrary irreducible polynomial. The multiplication result is computed after m clock cycles. The advantages of the proposed architecture are the high order of flexibility, which allows an easy configuration for variable field size 2^m, and the low hardware complexity, which results in small area. In addition, the proposed multiplier has low power consumption features, which are achieved by using the gated clock technique. Comparing with previous published implementations, the proposed multiplier architecture is suitable for devices with limited silicon area.

References

[1] S. Lin, D. Costello, Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[2] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1997.

[3] H. Li, C.N. Zhang, Efficient cellular automata based versatile multiplier for GF(2^m), J Inform. Sci. Engng. 18 (4) (2002) 479–488.

[4] Ç.K. Koç, B. Sunar, Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields, IEEE Trans. Comput. 47 (3) (1998) 353–356.

[5] C. Paar, P. Fleischmann, P. Soria-Rodriguez, Fast arithmetic for public-key algorithms in Galois field with composite exponents, IEEE Trans. Comput. 48 (10) (1999) 1025–1034.

[6] C. Paar, N. Lange, A comparative VLSI synthesis of finite field multipliers, Proceedings of the Third International Symposium of Communication Theory and Its Applications, Lake District, UK, July 1995.

[7] L. Song, K.K. Parhi, Low-energy digit-serial/parallel finite field multipliers, J. VLSI Signal Process. Syst. 19 (2) (1998) 149–166.

[8] G. Orlando, C. Paar, A scalable GF(p) elliptic curve processor architecture for programmable hardware, Proceedings of the Cryptographic Hardware and Embedded Systems—CHES, LNCS, vol. 2162, Springer, Paris, 2001, pp. 348–363.

[9] M.A. Hasan, M. Wang, V.K. Bhargava, Modular construction of low complexity parallel multipliers for a class of finite field GF(2^m), IEEE Trans. Comput. 41 (8) (1992) 962–971.

[10] H. Wu, Low complexity bit-parallel finite field arithmetic using polynomial basis, Proceedings of the cryptographic hardware and embedded systems—CHES, LNCS, vol. 1717, Springer, Worcester, MA, 1999, pp. 280–291.

[11] W. Geiselmann, H. Lukhaub, Redundant representation of finite fields, Proceedings of the Fourth International Workshop on Practice and Theory in Public Key Cryptosystems, PKC, Cheju Island, Korea, February 13–15, 2001.

[12] M.A. Hasan, M. Ebtedaei, Efficient architectures for computations over variable dimensional Galois field, IEEE Trans. Circuits Syst. I. Fundam. Theory Appl. 45 (11) (1998).

[13] E. Savas, A.F. Tenca, Ç.K. Koc, A scalable and unified multiplier architecture for finite field GF(p) and GF(2^m), Proceedings of the Cryptographic Hardware and Embedded Systems—CHES, LNCS, vol. 1965, Springer, Berlin, 2000, pp. 277–292.

[14] A.P. Chandrakasan, R.W. Brodersen, Low power digital CMOS design, Kluwer Academic Publishers, Dordrecht, 1995.

[15] N. Koblitz, Elliptic curve cryptosystems, Math. Comput. 48 (1987) 203–209.

[16] V. Miller, Uses of elliptic curves in cryptography, Proceedings of the Advances in Cryptology—CRYPTO'85, LNCS, vol. 218, Berlin, Germany, 1986, pp. 417–426.

[17] R. Lidl, H. Niederreiter, Introduction to finite fields and their applications, Cambridge University Press, Cambridge, 1986.

[18] P.A. Scott, S.E. Travares, L.E. Peppard, A fast VLSI multiplier for GF(2^m), IEEE J. Sel. Areas Commun. sac-4 (Jan) (1986) 62–65.

[19] S. Nandi, B.K. Kar, P.P. Chaudhuri, Theory and applications of cellular automata in cryptography, IEEE Trans. Comput. 43 (12) (1994) 1346–1357.

[20] N.P. Smart, How secure are elliptic curves over composite extension fields?, Proceedings of the Advances in Cryptology—EUROCRYPT, Innsbruck, Austria, May 6–10, 2001, pp. 30–39.

[21] Ç.K. Koc, T. Acar, Montgomery multiplication in GF(2^m), Des. Codes Cryptography 14 (1) (April 1998) 57–69.

[22] A.K. Lenstra, E.R. Verheulc, Selecting cryptographic key sizes, J. Cryptol. 14 (2) (2001) 255–293.

[23] Xilinx, Virtex, 2.5 V Field Programmable Gate Arrays, Xilinx, San Jose, CA, 2002, www.xilinx.com